Linear Algebra and its Applications 433 (2010) 1254–1264

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Polynomial least squares fitting in the Bernstein basis

Ana Marco, José-Javier Martínez *

Departamento de Matemáticas, Universidad de Alcalá, Campus Universitario, 28871 Alcalá de Henares, Madrid, Spain

ARTICLE INFO

Article history:
Received 3 July 2008
Accepted 7 June 2010
Available online 13 July 2010

Submitted by V. Mehrmann

AMS classification:
65F20
65F35
15A06
15A23

Keywords:
Least squares
Bernstein–Vandermonde matrix
Bernstein basis
Bidiagonal decomposition
Total positivity

ABSTRACT

The problem of polynomial least squares fitting in which the usual monomial basis is replaced by the Bernstein basis is considered. The coefficient matrix of the overdetermined system to be solved in the least squares sense is then a rectangular Bernstein–Vandermonde matrix. In order to use the method based on the QR decomposition of A, the first stage consists of computing the bidiagonal decomposition of the coefficient matrix A. Starting from that bidiagonal decomposition, an algorithm for obtaining the QR decomposition of A is then applied. Finally, a triangular system is solved by using the bidiagonal decomposition of the R-factor of A. Some numerical experiments showing the behavior of this approach are included.

1. Introduction

Given $\{x_i\}_{1 \leqslant i \leqslant l+1}$ pairwise distinct real points and $\{f_i\}_{1 \leqslant i \leqslant l+1} \in \mathbf{R}$, let us consider a degree n polynomial

$$P(x) = c_0 + c_1 x + \cdots + c_n x^n$$

for some $n \leqslant l$. Such a polynomial is a *least squares fit* to the data if it minimizes the sum of the squares of the deviations from the data,

* Corresponding author.
E-mail addresses: ana.marco@uah.es (A. Marco), jjavier.martinez@uah.es (J.-J. Martínez).

0024-3795/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2010.06.031

$$\sum_{i=1}^{l+1} |f_i - P(x_i)|^2.$$

Computing the coefficients c_j of that polynomial $P(x)$ is equivalent to solve, in the least squares sense, the overdetermined linear system $Ac = f$, where A is the rectangular $(l+1) \times (n+1)$ Vandermonde matrix corresponding to the nodes $\{x_i\}_{1 \leqslant i \leqslant l+1}$.

Taking into account that A has full rank $n+1$, the problem has a unique solution given by the unique solution of the linear system

$$A^T Ac = A^T f,$$

the *normal equations*.

Since A is usually an ill-conditioned matrix, it was early recognized that solving the normal equations was not an adequate method. Golub [8], following previous ideas by Householder, suggested the use of the *QR factorization* of A, which involves the solution of a linear system with the triangular matrix R.

Let us observe that, if $A = QR$ with Q being an orthogonal matrix, then using the condition number in the spectral norm we have

$$\kappa_2(R) = \kappa_2(A),$$

that is, R inherits the ill-conditioning of A while $\kappa_2(A^T A) = \kappa_2(A)^2$.

In addition, as it was already observed by Golub in [9] (see also Section 20.1 of [10]), *although the use of the orthogonal transformation avoids some of the ill effects inherent in the use of normal equations, the value $\kappa_2(A)^2$ is still relevant to some extent*.

Consequently a good idea is to use, instead of the monomial basis, a polynomial basis which leads to a matrix A with smaller condition number than the Vandermonde matrix.

It is frequently assumed that this happens when bases of orthogonal polynomials, such as the basis of Chebyshev polynomials, are considered. However, this fact is true when special sets of nodes are considered, but not in the case of general nodes.

As suggested by the experiments for the case of square Bernstein–Vandermonde matrices carried out in [14], a basis which leads to a matrix A better conditioned than the Vandermonde matrix is the Bernstein basis of polynomials, a widely used basis in Computer Aided Geometric Design due to the good properties that it possesses (see, for instance, [2,11]). We illustrate these facts with Table 1, where the condition numbers of Vandermonde, Chebyshev–Vandermonde and Bernstein–Vandermonde matrices are presented for the nodes considered in Examples 5.1 and 5.2.

Some theoretical results on the optimal conditioning of Bernstein–Vandermonde matrices along with several numerical examples have recently been presented by Delgado and Peña in [4].

Let us observe that, without the loss of generality, we can consider the nodes $\{x_i\}_{1 \leqslant i \leqslant l+1}$ ordered and belonging to $(0, 1)$. So, we will solve the following problem:

Given $\{x_i\}_{1 \leqslant i \leqslant l+1} \in (0, 1)$ a set of points such that $0 < x_1 < \cdots < x_{l+1} < 1$, compute a polynomial

$$P(x) = \sum_{j=0}^{n} c_j b_j^{(n)}(x)$$

expressed in the *Bernstein basis* of the space $\Pi_n(x)$ of the polynomials of degree less than or equal to n on the interval $[0, 1]$

Table 1
Condition numbers of the Vandermonde (V), Chebyshev–Vandermonde (TV) and Bernstein–Vandermonde (BV) matrices.

Example	V	TV	BV
5.1	1.7e+12	1.7e+12	2.0e+05
5.2	2.5e+14	4.0e+14	5.3e+08

$$\mathcal{B}_n = \left\{ b_j^{(n)}(x) = \binom{n}{j}(1-x)^{n-j}x^j,\ j = 0, \ldots, n \right\},$$

such that $P(x)$ minimizes the sum of the squares of the deviations from the data $\{f_i\}_{1 \leqslant i \leqslant l+1} \in \mathbf{R}$.

Solving this problem is equivalent to solve in the least squares sense the overdetermined linear system $Ac = f$, where now

$$A = \begin{pmatrix} \binom{n}{0}(1-x_1)^n & \binom{n}{1}x_1(1-x_1)^{n-1} & \cdots & \binom{n}{n}x_1^n \\ \binom{n}{0}(1-x_2)^n & \binom{n}{1}x_2(1-x_2)^{n-1} & \cdots & \binom{n}{n}x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \binom{n}{0}(1-x_{l+1})^n & \binom{n}{1}x_{l+1}(1-x_{l+1})^{n-1} & \cdots & \binom{n}{n}x_{l+1}^n \end{pmatrix} \tag{1.1}$$

is the $(l+1) \times (n+1)$ Bernstein–Vandermonde matrix for the Bernstein basis $\mathcal{B}_n$ and the nodes $\{x_i\}_{1 \leqslant i \leqslant l+1}$,

$$f = (f_1, f_2, \ldots, f_{l+1})^T \tag{1.2}$$

is the data vector, and

$$c = (c_1, c_2, \ldots, c_{n+1})^T \tag{1.3}$$

is the vector containing the coefficients of the polynomial that we want to compute.

Taking into account that Bernstein–Vandermonde matrices are strictly totally positive matrices when the nodes satisfy $0 < x_1 < x_2 < \cdots < x_{l+1} < 1$ (see [2]), we will use results from the field of totally positive matrices (mainly Neville elimination) for solving our polynomial regression problem.

The rest of the paper is organized as follows. Total positivity and Neville elimination are considered in Section 2. In Section 3, the bidiagonal factorization of a rectangular Bernstein–Vandermonde matrix is presented. The algorithm for computing the regression polynomial in Bernstein basis is given in Section 4. Finally, Section 5 is devoted to illustrate the accuracy of our algorithm by means of some numerical experiments.

2. Basic results on total positivity and Neville elimination

In this section we will briefly recall some basic results on total positivity and Neville elimination which we will apply in Section 3. Our notation follows the notation used in [5,6]. Given k, $n \in \mathbf{N}$ $(1 \leqslant k \leqslant n)$, $Q_{k,n}$ will denote the set of all increasing sequences of k positive integers less than or equal to n.

Let A be an $l \times n$ real matrix. For $k \leqslant l$, $m \leqslant n$, and for any $\alpha \in Q_{k,l}$ and $\beta \in Q_{m,n}$, we will denote by $A[\alpha|\beta]$ the submatrix $k \times m$ of A containing the rows numbered by $\alpha$and the columns numbered by β.

The fundamental tool for obtaining the results presented in this paper is the *Neville elimination* [5,6], a procedure that makes zeros in a matrix adding to a given row an appropriate multiple of the previous one. We will describe the Neville elimination for a matrix $A = (a_{i,j})_{1 \leqslant i \leqslant l; 1 \leqslant j \leqslant n}$ where $l \geqslant n$.

Let $A = (a_{i,j})_{1 \leqslant i \leqslant l; 1 \leqslant j \leqslant n}$ be a matrix where $l \geqslant n$. The Neville elimination of A consists of $n-1$ steps resulting in a sequence of matrices $A := A_1 \to A_2 \to \cdots \to A_n$, where $A_t = (a_{i,j}^{(t)})_{1 \leqslant i \leqslant l; 1 \leqslant j \leqslant n}$ has zeros below its main diagonal in the $t-1$ first columns. The matrix A_{t+1} is obtained from A_t $(t = 1, \ldots, n)$ by using the following formula:

$$a_{i,j}^{(t+1)} := \begin{cases} a_{i,j}^{(t)}, & \text{if } i \leqslant t, \\ a_{i,j}^{(t)} - \left(a_{i,t}^{(t)}/a_{i-1,t}^{t}\right) a_{i-1,j}^{(t)}, & \text{if } i \geqslant t+1 \text{ and } j \geqslant t+1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

In this process the element

$$p_{i,j} := a_{i,j}^{(j)}, \quad 1 \leqslant j \leqslant n,\ j \leqslant i \leqslant l$$

is called *pivot* (i,j) of the Neville elimination of A. The process would break down if any of the pivots $p_{i,j}(1 \leqslant j \leqslant n, j \leqslant i \leqslant l)$ is zero. In that case we can move the corresponding rows to the bottom and proceed with the new matrix, as described in [5]. The Neville elimination can be done without row exchanges if all the pivots are nonzero, as it will happen in our situation. The pivots $p_{i,i}$ are called *diagonal pivots*. If all the pivots $p_{i,j}$ are nonzero, then $p_{i,1} = a_{i,1}\ \forall i$ and, by Lemma 2.6 of [5]

$$p_{i,j} = \frac{\det A[i-j+1,\ldots,i|1,\ldots,j]}{\det A[i-j+1,\ldots,i-1|1,\ldots,j-1]}, \quad 1 < j \leqslant n,\ j \leqslant i \leqslant l. \tag{2.2}$$

The element

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}}, \quad 1 \leqslant j \leqslant n,\ j < i \leqslant l \tag{2.3}$$

is called *multiplier* of the Neville elimination of A. The matrix $U := A_n$ is upper triangular and has the diagonal pivots in its main diagonal.

The *complete Neville elimination* of a matrix A consists in performing the Neville elimination of A for obtaining U and then continuing with the Neville elimination of U^T. The pivot (respectively, multiplier) (i,j) of the complete Neville elimination of A is the pivot (respectively, multiplier) (j,i) of the Neville elimination of U^T, if $j \geqslant i$. When no row exchanges are needed in the Neville elimination of A and U^T, we say that the complete Neville elimination of A can be done without row and column exchanges, and in this case the multipliers of the complete Neville elimination of A are the multipliers of the Neville elimination of A if $i \geqslant j$ and the multipliers of the Neville elimination of A^T if $j \geqslant i$.

A matrix is called *totally positive* (respectively, *strictly totally positive*) if all its minors are nonnegative (respectively, positive). The Neville elimination characterizes the strictly totally positive matrices as follows [5]:

Theorem 2.1. *A matrix is strictly totally positive if and only if its complete Neville elimination can be performed without row and column exchanges, the multipliers of the Neville elimination of A and A^T are positive, and the diagonal pivots of the Neville elimination of A are positive.*

It is well known [2] that the Bernstein–Vandermonde matrix is a strictly totally positive matrix when the nodes satisfy $0 < x_1 < x_2 < \cdots < x_{l+1} < 1$, but this result is also a consequence of our Theorem 3.2.

3. Bidiagonal factorization of *A*

In this section we consider the bidiagonal factorization of the Bernstein–Vandermonde matrix A of (1.1), which, as we shall see, will be the starting point for the computation of the QR decomposition of A.

Let us observe that when $l = n$ this matrix A is the coefficient matrix of the linear system associated with a Lagrange interpolation problem in the Bernstein basis $\mathcal{B}_n$ whose interpolation nodes are $\{x_i : i = 1,\ldots,n+1\}$. A fast and accurate algorithm for solving this linear system, and therefore the corresponding Lagrange interpolation problem in the Bernstein basis can be found in [14]. A good introduction to the interpolation theory can be seen in [3].

The following result will be very important in the construction of our algorithm, since by (2.2) and (2.3) the multipliers and the diagonal pivots of Neville elimination can be expressed in terms of certain determinants.

Proposition 3.1 (see [14])**.** *Let A be the square Bernstein–Vandermonde matrix of order $n+1$ for the Bernstein basis $\mathcal{B}_n$ and the nodes $x_1, x_2, \ldots, x_{n+1}$. We have:*

$$\det A = \binom{n}{0}\binom{n}{1}\cdots\binom{n}{n}\prod_{1\leqslant i<j\leqslant n+1}(x_j - x_i).$$

When A is a totally positive matrix, the process of Neville elimination allows us to obtain a bidiagonal factorization of A [7, 13] as shown in the next theorem, which is a crucial result for the construction of our algorithm.

Theorem 3.2. *Let $A = (a_{i,j})_{1\leqslant i\leqslant l+1;1\leqslant j\leqslant n+1}$ be a Bernstein–Vandermonde matrix for the Bernstein basis $\mathcal{B}_n$ whose nodes satisfy $0 < x_1 < x_2 < \cdots < x_l < x_{l+1} < 1$. Then A admits a factorization in the form*

$$A = F_l F_{l-1}\cdots F_1 D G_1 \cdots G_{n-1} G_n \tag{3.1}$$

where $F_i (1\leqslant i\leqslant l)$ are $(l+1)\times(l+1)$ bidiagonal matrices of the form

$$F_i = \begin{pmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & 1 & & & & \\ & & & m_{i+1,1} & 1 & & & \\ & & & & m_{i+2,2} & 1 & & \\ & & & & & \ddots & \ddots & \\ & & & & & & m_{l,l-i} & 1 \end{pmatrix}, \tag{3.2}$$

$G_i^T (1\leqslant i\leqslant n)$ are $(n+1)\times(n+1)$ bidiagonal matrices of the form

$$G_i^T = \begin{pmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & 1 & & & & \\ & & & \tilde{m}_{i+1,1} & 1 & & & \\ & & & & \tilde{m}_{i+2,2} & 1 & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \tilde{m}_{n,n-i} & 1 \end{pmatrix}, \tag{3.3}$$

and D is the $(l+1)\times(n+1)$ diagonal matrix whose ith $(1\leqslant i\leqslant n+1)$ diagonal entry is the diagonal pivot $p_{i,i} = a_{i,i}^{(i)}$ of the Neville elimination of A:

$$D = (d_{i,j})_{1\leqslant i\leqslant l+1;1\leqslant j\leqslant n+1} = \mathrm{diag}\{p_{1,1}, p_{2,2}, \ldots, p_{n+1,n+1}\}. \tag{3.4}$$

In addition, the entries of those bidiagonal and diagonal matrices (i.e. the multipliers $m_{i,j}$ of the Neville elimination of A, the multipliers $\tilde{m}_{i,j}$ of the Neville elimination of A^T, and the diagonal pivots $p_{i,i}$) are given by

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}} = \frac{(1-x_i)^{n-j+1}(1-x_{i-j})\prod_{k=1}^{j-1}(x_i - x_{i-k})}{(1-x_{i-1})^{n-j+2}\prod_{k=2}^{j}(x_{i-1} - x_{i-k})}$$

$(j = 1, \ldots, n+1;\ i = j+1, \ldots, l+1),$

$$\tilde{m}_{i,j} = \frac{(n-i+2)\cdot x_j}{(i-1)(1-x_j)}, \quad j = 1, \ldots, n; \quad i = j+1, \ldots, n+1$$

and

$$p_{i,i} = \frac{\binom{n}{i-1}(1-x_i)^{n-i+1}\prod_{k<i}(x_i - x_k)}{\prod_{k=1}^{i-1}(1-x_k)}, \quad i = 1, \ldots, n+1.$$

When in the expression of F_i we have some multiplier $m_{k,j}$ which does not exist (because $j > n+1$), then the corresponding entry is equal to zero, and analogously for the entries in G_i^T.

Proof. The matrix A is strictly totally positive (see [2]) and therefore, by Theorem 2.1, the complete Neville elimination of A can be performed without row and column exchanges providing the factorization of A given in the statement of the theorem (see [7,13]):

$$A = F_l F_{l-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n.$$

Taking into account that the minors of A with j initial consecutive columns and j consecutive rows starting with row i are

$$\det A[i,\ldots,i+j-1|1,\ldots,j] = \binom{n}{0}\binom{n}{1}\cdots\binom{n}{j-1}$$

$$(1-x_i)^{n-j+1}(1-x_{i+1})^{n-j+1}\cdots(1-x_{i+j-1})^{n-j+1}\prod_{i\leqslant k<h\leqslant i+j-1}(x_h-x_k),$$

a result that follows from the properties of the determinants and Proposition 3.1, and that $m_{i,j}$ are the multipliers of the Neville elimination of A, we obtain that

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}} = \frac{(1-x_i)^{n-j+1}(1-x_{i-j})\prod_{k=1}^{j-1}(x_i-x_{i-k})}{(1-x_{i-1})^{n-j+2}\prod_{k=2}^{j}(x_{i-1}-x_{i-k})}, \tag{3.5}$$

where $j = 1,\ldots,n+1$ and $i = j+1,\ldots,l+1$.

As for the minors of A^T with j initial consecutive columns and j consecutive rows starting with row i, they are:

$$\det A^T[i,\ldots,i+j-1|1,\ldots,j] = \binom{n}{i-1}\binom{n}{i}\cdots\binom{n}{i+j-2}x_1^{i-1}x_2^{i-1}\cdots x_j^{i-1}$$

$$(1-x_1)^{n-i-j+2}(1-x_2)^{n-i-j+2}\cdots(1-x_j)^{n-i-j+2}\prod_{1\leqslant k<h\leqslant j}(x_h-x_k).$$

This expression also follows from the properties of the determinants and Proposition 3.1. Since the entries $\tilde{m}_{i,j}$ are the multipliers of the Neville elimination of A^T, using the previous expression for the minors of A^T with initial consecutive columns and consecutive rows, it is obtained that

$$\tilde{m}_{i,j} = \frac{(n-i+2)\cdot x_j}{(i-1)(1-x_j)}, \quad j = 1,\ldots,n; \quad i = j+1,\ldots,n+1. \tag{3.6}$$

Finally, the ith diagonal entry of D

$$p_{i,i} = \frac{\binom{n}{i-1}(1-x_i)^{n-i+1}\prod_{k<i}(x_i-x_k)}{\prod_{k=1}^{i-1}(1-x_k)}, \quad i = 1,\ldots,n+1 \tag{3.7}$$

is obtained by using the expression for the minors of A with initial consecutive columns and initial consecutive rows. □

Moreover, by using similar arguments to those used in [6] it can be seen that this factorization is unique among factorizations of this type, that is to say, factorizations in which the matrices involved have the properties shown by formulae (3.2)–(3.4).

The bidiagonal decomposition of Theorem 3.2 can be expressed in a compact way by storing the multipliers and the diagonal pivots in a matrix which has been called $\mathcal{BD}(A)$ by Koev [13].

For example, for the 4×2 Bernstein–Vandermonde matrix

$$A = \begin{pmatrix} 4/5 & 1/5 \\ 3/5 & 2/5 \\ 2/5 & 3/5 \\ 1/5 & 4/5 \end{pmatrix}$$

its bidiagonal factorization is

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2/3 & 1 & 0 \\ 0 & 0 & 3/2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3/4 & 1 & 0 & 0 \\ 0 & 4/3 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$\times \begin{pmatrix} 4/5 & 0 \\ 0 & 1/4 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1/4 \\ 0 & 1 \end{pmatrix}$$

and the corresponding $\mathcal{BD}(A)$ is

$$\mathcal{BD}(A) = \begin{pmatrix} 4/5 & 1/4 \\ 3/4 & 1/4 \\ 2/3 & 4/3 \\ 1/2 & 3/2 \end{pmatrix}.$$

Now it must be stressed that while Neville elimination has been the key theoretical tool for the analysis of the bidiagonal decomposition of A, it generally fails to provide an accurate algorithm for computing $\mathcal{BD}(A)$. This fact is explicitly noted in [12], where the author indicated that *the function TNBD is the only function in the package TNTool that does not guarantee high relative accuracy*.

Consequently, the accurate computation of $\mathcal{BD}(A)$ is an important task which will be addressed in Section 4, the explicit expressions are given by (3.5)–(3.7) being crucial for obtaining high relative accuracy.

Remark 1. The formulae obtained in the proof of Theorem 3.3 for the minors of A with j initial consecutive columns and j consecutive rows, and for the minors of A^T with j initial consecutive columns and j consecutive rows show that they are not zero and so, the complete Neville elimination of A can be performed without row and column exchanges. Looking at Eqs. (3.5)–(3.7) it is easily seen that $m_{i,j}$, $\tilde{m}_{i,j}$ and $p_{i,i}$ are positive. Therefore, taking into account Theorem 2.1, this confirms that the matrix A is strictly totally positive.

Remark 2. In the square case, the matrices F_i $(i = 1, \ldots, l)$ and the matrices G_j $(j = 1, \ldots, n)$ are not the same bidiagonal matrices that appear in the bidiagonal factorization of A^{-1} presented in [14], nor their inverses. The multipliers of the Neville elimination of A and A^T give us the bidiagonal factorization of A and A^{-1}, but obtaining the bidiagonal factorization of A from the bidiagonal factorization of A^{-1} (or vice versa) is not straightforward, since the structure of the bidiagonal matrices that appear in both factorizations is not preserved by the inversion, that is, in general, $F_i^{-1}(i = 1, \ldots, l)$ and $G_j^{-1}(j = 1, \ldots, n)$ are not bidiagonal matrices. We illustrate this fact by means of the following example with the Vandermonde matrix

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 1 & 5 & 25 \\ 1 & 9 & 81 \end{pmatrix}.$$

We have the following bidiagonal decomposition of A:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 4/3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 28 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

while for the inverse of A we have the factorization

$$A^{-1}=\begin{pmatrix}1 & -2 & 0\\ 0 & 1 & -2\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & -5\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 0 & 1/3 & 0\\ 0 & 0 & 1/28\end{pmatrix}$$
$$\times\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -4/3 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ -1 & 1 & 0\\ 0 & -1 & 1\end{pmatrix}$$

Nevertheless, it is important to note that

$$\mathcal{BD}(A)=\begin{pmatrix}1 & 2 & 2\\ 1 & 3 & 5\\ 1 & 4/3 & 28\end{pmatrix}$$

serves to represent *both* factorizations.

4. The algorithm

In this section we present an accurate and efficient algorithm for solving the problem of polynomial regression in Bernstein basis we have presented in Section 1. As we introduced there, our algorithm is based on the solution of the least squares problem $min_c\|f-Ac\|_2$, where A, f and c are given by (1.1)–(1.3), respectively. Taking into account that A is a strictly totally positive matrix, it has full rank, and the method based on the QR decomposition is adequate for solving this least squares problem [1].

The following result (see Section 1.3.1 in [1]) will be essential in the construction of our algorithm.

Theorem 4.1. *Let $Ac=f$ a linear system where $A\in\mathbf{R}^{(l+1)\times(n+1)}$, $l\geqslant n$, $c\in\mathbf{R}^{n+1}$ and $f\in\mathbf{R}^{l+1}$. Assume that $\operatorname{rank}(A)=n+1$, and let the QR decomposition of A be given by*

$$A=Q\begin{pmatrix}R\\ 0\end{pmatrix},$$

where $Q\in\mathbf{R}^{(l+1)\times(l+1)}$ is an orthogonal matrix and $R\in\mathbf{R}^{(n+1)\times(n+1)}$ is an upper triangular matrix with positive diagonal entries. Then the solution of the least squares problem $min_c\|f-Ac\|_2$ is obtained from

$$\begin{pmatrix}d_1\\ d_2\end{pmatrix}=Q^Tf,\quad Rc=d_1,\quad r=Q\begin{pmatrix}0\\ d_2\end{pmatrix},$$

where $d_1\in\mathbf{R}^{n+1}$, $d_2\in\mathbf{R}^{l-n}$ and $r=f-Ac$. In particular $\|r\|_2=\|d_2\|_2$.

An accurate and efficient algorithm for computing the QR decomposition of a strictly totally positive matrix A is presented in [13]. This algorithm is called `TNQR` and can be obtained from [12]. Given the bidiagonal factorization of A, `TNQR` computes the matrix Q and the bidiagonal factorization of the matrix R. Let us point out here that if A is strictly totally positive, then R is strictly totally positive. `TNQR` is based on Givens rotations, has a computational cost of $O(l^2n)$ arithmetic operations if the matrix Q is required, and its high relative accuracy comes from the avoidance of subtractive cancellation.

A fast and accurate algorithm for computing the bidiagonal factorization of the rectangular Bernstein–Vandermonde matrix that appears in our problem of polynomial regression in the Bernstein basis can be developed by using the expressions (3.5)–(3.7) for the computation of the multipliers $m_{i,j}$ and $\tilde{m}_{i,j}$, and the diagonal pivots $p_{i,i}$ of its Neville elimination. The algorithm is an extension to the rectangular case of the one presented in [14] for the square Bernstein–Vandermonde matrices. Given the nodes $\{x_i\}_{1\leqslant i\leqslant l+1}\in(0,1)$ and the degree n of the Bernstein basis, it returns a matrix $M\in\mathbf{R}^{(l+1)\times(n+1)}$ such that

$$M_{i,i} = p_{i,i}, \quad i = 1, \ldots, n+1,$$
$$M_{i,j} = m_{i,j}, \quad j = 1, \ldots, n+1; \quad i = j+1, \ldots, l+1,$$
$$M_{i,j} = \tilde{m}_{j,i}, \quad i = 1, \ldots, n; \quad j = i+1, \ldots, n+1.$$

The algorithm, which we call TNBDBV, has a computational cost of $O(ln)$ arithmetic operations, and high relative accuracy because it only involves arithmetic operations that avoid subtractive cancellation (see [14] for the details). The implementation in MATLAB of the algorithm in the square case can be taken from [12].

In this way, the algorithm for solving the least squares problem $min_c\|f - Ac\|_2$ corresponding to our polynomial regression problem will be:

INPUT: The nodes $\{x_i\}_{1 \leqslant i \leqslant l+1} \in (0, 1)$, the data vector f and the degree n of the Bernstein basis.
OUTPUT: A vector $c = (c_j)_{1 \leqslant i \leqslant n+1}$ containing the coefficients of the polynomial $P(x)$ in the Bernstein basis $\mathcal{B}_n$ and the minimum residual r.

- *Step 1*: Computation of the bidiagonal factorization of A by means of TNBDBV.
- *Step 2*: Given the matrix M obtained in Step 1, computation of the QR decomposition of A by using TNQR.
- *Step 3*: Computation of

$$d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = Q^T f.$$

- *Step 4*: Solution of the upper triangular system $Rc = d_1$.
- *Step 5*: Computation of

$$r = Q \begin{pmatrix} 0 \\ d_2 \end{pmatrix}.$$

Steps 3 and 5 are carried out by using the standard matrix multiplication command of MATLAB. As for Step 4, it is done by means of the algorithm `TNSolve` of Koev [12]. Given the bidiagonal factorization of the totally positive matrix R, `TNSolve` solves a linear system whose coefficient matrix is R by using backward substitution.

Let us observe that A is not constructed, although we are also computing the residual $r = f - Ac$.

5. Numerical experiments and final remarks

Two numerical experiments illustrating the good properties of our algorithm are reported in this section. We solve the least squares problem $min_c\|f - Ac\|_2$ corresponding to the computation of the regression polynomial in exact arithmetic by means of the command `leastsqrs` of *Maple 10* and we denote this solution by c_e. We also compute the minimum residual r_e in exact arithmetic by using *Maple 10*. We use c_e and r_e for comparing the accuracy of the results obtained in MATLAB by means of:

(1) The algorithm presented in Section 4.
(2) The command $A\backslash f$ of MATLAB.

The relative errors obtained when using the approaches (1) and (2) for computing the coefficients of the regression polynomial in the experiments described in this section (ec_1 and ec_2, respectively) are included in the first and the third columns of Table 2. The relative errors corresponding to the computation of the minimum residual by using the approaches (1) and (2) (er_1 and er_2, respectively) are presented in the second and the fourth columns of Table 2.

We compute the relative error of the solution c of the least squares problem $min_c\|f - Ac\|_2$ by means of the formula

$$ec = \frac{\|c - c_e\|_2}{\|c_e\|_2}.$$

Table 2
Relative errors in Examples 5.1 and 5.2.

Example	ec_1	er_1	ec_2	er_2
5.1	1.4e−15	1.3e−15	7.8e−12	5.1e−12
5.2	2.0e−15	2.3e−15	1.4e−09	1.5e−08

The relative error of the minimum residual r is computed by means of

$$er = \frac{\|r - r_e\|_2}{\|r_e\|_2}.$$

Example 5.1. Let $\mathcal{B}_{15}$ the Bernstein basis of the space of polynomials with degree less than or equal to 15 in [0, 1]. We will compute the polynomial

$$P(x) = \sum_{j=0}^{15} c_j b_j^{(n)}(x)$$

that minimizes

$$\sum_{i=1}^{21} |f_i - P(x_i)|^2,$$

where

$$\{x_i\}_{1 \leqslant i \leqslant 21} = \left\{ \frac{i}{22} \right\}_{1 \leqslant i \leqslant 21},$$

and

$$f = (3, 4, 0, -2, 5, 0, 1, 9, -3, 7, -1, 0, 2, 2, -4, -2, 3, 8, -6, 4, 1)^T.$$

The condition number of the Bernstein–Vandermonde matrix A of the least squares problem corresponding to the regression polynomial we are interested in computing is $\kappa_2(A) = 2.0e + 05$.

The following example shows how the algorithm we have presented in this paper keeps the accuracy when the condition number of the Bernstein–Vander-monde matrix involved in the regression problem increases, while the accuracy of the general approach (2) which does not exploit the structure of this matrix goes down.

Example 5.2. We consider a regression problem such that the Bernstein basis $\mathcal{B}_{15}$ and the data vector f are the same as in Example 5.1. The points $\{x_i\}_{1 \leqslant i \leqslant 21}$ are now:

$$\left\{ \frac{1}{22}, \frac{1}{20}, \frac{1}{18}, \frac{1}{16}, \frac{1}{14}, \frac{1}{12}, \frac{1}{10}, \frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{2}, \frac{23}{42}, \frac{21}{38}, \frac{19}{34}, \frac{17}{30}, \frac{15}{26}, \frac{13}{22}, \frac{11}{18}, \frac{9}{14}, \frac{7}{10}, \frac{5}{6} \right\}.$$

The condition number of the Bernstein–Vandermonde matrix A involved in this experiment is $\kappa_2(A) = 5.3e + 08$.

Remark 3. The accuracy of our algorithm is obtained by exploiting the structure of the Bernstein–Vandermonde matrix. Every step of our algorithm, except the ones in which the standard matrix multiplication command of Matlab is used, is developed with high relative accuracy because only arithmetic operations that avoid subtractive cancellation are involved [13,14].

Remark 4. Our algorithm has the same computational cost ($O(l^2 n)$ arithmetic operations) as that of the conventional algorithms that solve the least squares problem by means of the QR decomposition ignoring the structure of the matrix, when Q is explicitly required (see Section 2.4.1 of [1]).

Acknowledgements

The authors are grateful to an anonymous referee whose suggestions have greatly helped to improve the manuscript. This research has been partially supported by Research Grant MTM2009-07315 from the Spanish Government.

References

[1] A. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.
[2] J.M. Carnicer, J.M. Peña, Shape preserving representations and optimality of the Bernstein basis, Adv. Comput. Math. 1 (1993) 173–196.
[3] P.J. Davis, Interpolation and Approximation, Dover Publications Inc., New York, 1975.
[4] J. Delgado, J.M. Peña, Optimal conditioning of Bernstein collocation matrices, SIAM J. Matrix Anal. Appl. 31 (2009) 990–996.
[5] M. Gasca, J.M. Peña, Total positivity and Neville elimination, Linear Algebra Appl. 165 (1992) 25–44.
[6] M. Gasca, J.M. Peña, A matricial description of Neville elimination with applications to total positivity, Linear Algebra Appl. 202 (1994) 33–45.
[7] M. Gasca, J.M. Peña, On factorizations of totally positive matrices, in: M. Gasca, C.A. Michelli (Eds.), Total Positivity and Its Applications, Kluwer Academic Publishers, Dordrecht, 1996, pp. 109–130.
[8] G.H. Golub, Numerical methods for solving linear least squares problems, Numer. Math. 7 (1965) 206–216.
[9] G.H. Golub, Matrix decompositions and statistical calculations, in: R.C. Milton, J.A. Nedler (Eds.), Statistical Computations, Academic Press, New York, 1969, pp. 365–397.
[10] N.J. Higham, Accuracy and Stability of Numerical Algorithms, second ed., SIAM, Philadelphia, 2002.
[11] J. Hoschek, D. Lasser, Fundamentals of Computer Aided Geometric Design, A.K. Peters, Wellesley, 1993.
[12] P. Koev, 2007, <http://math.mit.edu/∼plamen/software/TNTool.html>.
[13] P. Koev, Accurate computations with totally nonnegative matrices, SIAM J. Matrix Anal. Appl. 29 (2007) 731–751.
[14] A. Marco, J.J. Martínez, A fast and accurate algorithm for solving Bernstein–Vandermonde linear systems, Linear Algebra Appl. 422 (2007) 616–628.